ABSTRACT OF THE DISCLOSURE

A METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR PROVIDING EFFICIENT UTILIZATION OF MEMORY HIERARCHY THROUGH CODE RESTRUCTURING

Code restructuring or reordering based on profiling information and memory hierarchy is provided by constructing a *Program Execution Graph* (PEG) corresponding to a level of the memory hierarchy, partitioning this PEG to reduce estimated memory overhead costs below an upper bound, and constructing a PEG for a next level of the memory hierarchy from the partitioned PEG. The PEG is constructed from control flow and frequency information from a profile of the program to be restructured. The PEG is a weighted undirected graph comprising nodes representing basic blocks and edges representing transfer of control between pairs of basic blocks. The weight of a node is the size of the basic block it represents and the weight of an edge is the frequency of transition between the pair of basic blocks it connects. The nodes of the PEG are partitioned or clustered into clusters such that the sum of the weights of the nodes in any cluster is no greater than an upper bound. A next PEG is then constructed from the clusters of the partitioned PEG such that a node in the next PEG corresponds to a cluster in the partitioned PEG, and such that there is an edge between two nodes in the next PEG if there is an edge between the clusters represented by the two nodes. Weights are assigned to the nodes and edges of the next PEG to produce a PEG, and then the PEG partitioning, basic block reordering, and PEG construction steps may be repeated for each level of the memory hierarchy. After the clustering is completed, the basic blocks are reordered in memory by grouping all of the nodes of a cluster in an adjacent order beginning at a boundary for all the levels of the memory hierarchy. Because clusters must not cross boundaries of memory hierarchies, NOPs are added to fill out the portion of a memory hierarchy level that is not filled by the clusters.